CLAIMS

What is claimed is:

1. A computer implemented method of analyzing multi-threaded programs, comprising:

suspending a first thread that requests a synchronization object that could result in a deadlock if acquired, the deadlock being evidenced by if another thread previously held the synchronization object while acquiring another synchronization object;

receiving a request from a second thread to acquire the synchronization object while the first thread is suspended;

allowing the second thread to acquire the synchronization object; and

awakening the first thread to potentially produce a deadlock.

2. The method of claim 1, further comprising checking whether the first and second thread are deadlocked by the first thread waiting to acquire a synchronization object that the second thread holds and the second thread waiting to acquire a synchronization object that the first thread holds.

3. The method of claim 1, wherein the first thread is suspended for a predetermined time, meaning that the first thread awakens after the predetermined time expires.

4. The method of claim 3, wherein the thread is also suspended on an event, meaning that the event awakens the first thread.

5. The method of claim 4, wherein the second thread sends the event that awakens the first thread.

6. The method of claim 1, wherein the first and second threads can hold a plurality of synchronization objects at a time.

7. The method of claim 1, wherein only one thread can hold the synchronization object at a time.

8. The method of claim 1, wherein only the first and second threads can release synchronization objects that each holds.

9. A computer program product for analyzing multi-threaded programs, comprising:

computer code that suspends a first thread that requests a synchronization object that could result in a deadlock if acquired, the deadlock being evidenced by if another thread previously held the synchronization object while acquiring another synchronization object;

computer code that determines if another thread previously held the synchronization object while acquiring another synchronization object;

computer code that suspends the first thread if another thread previously held the synchronization object while acquiring another synchronization object;

computer code that receives a request from a second thread to acquire the synchronization object while the first thread is suspended;

computer code that allows the second thread to acquire the synchronization object;

computer code that awakens the first thread to potentially produce a deadlock; and

a computer readable medium that stores the computer codes.

10. The computer program product of claim 9, wherein the computer readable medium is selected from the group consisting of CD-ROM, floppy disk, tape, flash memory, system memory, hard drive, and data signal embodied in a carrier wave.